Localization Using Cellular Infrastructure

Submitted By

Bhimani Yogesh

13MCEI04

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Localization Using Cellular Infrastructure

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

(Information and Network Security)

Submitted By

Bhimani Yogesh

(13MCEI04)

Guided By

Prof. Manish Chaturvedi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Certificate

This is to certify that the major project entitled **"Localization Using Cellular Infrastructure"** submitted by **Bhimani Yogesh (Roll No: 13MCEI04)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Institute of Technology,Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Manish Chaturvedi
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sharda Valeveti
Associate Professor,
Coordinator M.Tech - INS
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Bhimani Yogesh**, Roll. No.**13MCEI04**, give undertaking that the Major Project entitled **"Localization Using Cellular Infrastructure"** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by

Prof. Manish Chaturvedi

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Manish Chaturvedi**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

See that you acknowledge each one who have helped you in the project directly or indirectly.

- Bhimani Yogesh
13MCEI04

Abstract

Global Positioning System(GPS) is constantly considered as a best localization method. GPS technology utilizes numerous satellites for discovering accurate area,But there are some disadvantages of GPS technology. GPS does not work in indoor situation and in comparable ranges with restricted view-able pathway with GPS satellite. An alternate issue with GPS is particular equipment expected to join with satellite,which increase the expense. In our task we are attempting to get option of GPS localization. We are trying to use GSM network to find the location. GSM signals will work even in indoor circumstance. GPS gadgets consumes more battery as contrast with GSM cell telephones. As per TRAI Press Release No. 32/2015, there are more than 970.97 million cellular connections in India and cellular density is reported to be more than 77.58%. Aim of our work is to study feasibility of using cellular infrastructure for localization

In this project we are utilizing GSM fingerprinting algorithm which lives up to expectations in two stage learning stage and prediction stage. In learning stage we are collecting GSM traces for GSM network. We also collect GPS traces corresponding to the GSM traces. For collecting the traces we made an android application. Using this application we can record the GSM and GPS traces. This traces are used for training data and using this traces we will create the learning database. In prediction stage we used training database for predict the location. Using this algorithms we are getting a good accuracy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Definition

GPS(Global Positioning System)is the good option to track any devices or Gadgets. GPS has some issues as it adds additional expense to the gadget and its poor scope inside the building. The best possible option we can utilize is cellular network which give better scope and good accuracy. Cellular network i.e. GSM network is well established network for localization. GSM signal is also available within the building. We are trying to implement new scheme which can be used as alternative of GPS and for that use of GSM network is good option.

1.2 Motivation

GPS technology is well established localization technique but there are few issues with GPS technology. When we continually utilize GPS main issue is battery it consumes. GPS equipment consume more power and hence expends more battery furthermore GPS doesn't work where observable pathway is not accessible with the satellite. Thus it will not provide good results for indoor scenarios. As per TRAI Press Release No. 32/2015, there are more than 970.97 million cellular connections in India and cellular density is reported to be more than 77.58% [1]. All these issues motivated us to discover an option approach where we can conquer these issue and also this days multiple sim phones are available so using data of multiple sims we can implement new localization technique.

1.3 Advantages Of GSM Localization

- GSM network is a stable network therefore requires significant installation cost.
- It is generally managed network,meaning no obstruction from high exhausting access directs that are tuned toward the same channel.
- It is work in authorized band.Thus no obstruction from microwave stoves and other electronic supplies.
- Coverage is greater than other communication network.

1.4 Scope

Option of the GPS technology for localization is the Cellular network. GSM signs functions admirably for indoor scenario and for outdoor localization it will be an extra to the GPS technology. We will no more rely on upon GPS for localization. GSM restriction can be utilized anyplace whenever yet former learning database is required,therefor we can not utilize this plan without learning database.

Chapter 2

Literature Survey

2.1 Current Techniques

There are other technology for localization is also available other than GPS and GSM.These techniques are Infrared,ultrasound,Wi-Fi and Bluetooth localization.These techniques are either costly or not technically possible.Distinction between Wifi and GSM system is the scope of GSM cell can be up to 35km, which is 70 times higher than Wifi's most extreme scope of around 500m furthermore the organization of GSM systems is steady and arranged contrasted with the more ad hoc arrangement of Wifi access focuses.[2]

Trevisani et al. in [3] and Raja et al. in [4] study performance of various location technologies used in GSM cellular network and report that Enhanced Observed Time Difference (E-OTD) has location error of 50-150 meters independent of cell size.M.Y.Chen et al. in [5] compare performance of Centroid based, Finger printing based and Gaussian Process based localization algorithm, for Downtown area (66 cells per square kilometers) and Residential area (26 cells per square kilometers) with respect to location error, amount of training required, speed of execution, and storage requirement. They carried out performance analysis using top seven in-range cells from single/all providers, and all in-range cells from all providers. They reported the location errors as mentioned in table 2.1. Varshavsky et al. in [6] used Fingerprinting and Centroid algorithm for indoor and outdoor localization.They report median error of 75 meters and 213 meters with fingerprinting and Centroid algorithm respectively.

Valerio et al. [7] analyze mobile hand off related cellular network signaling data to generate road traffic information. They make following observations (1) cellular signaling

Table 2.1: Performance of Positioning Algorithms: Median Location Error (Meters)

Algorithm	**Downtown (66 cells/sqkm)**			**Residential(26 cells/sqkm)**		
	Single Provider (7)	Cross Provider (7)	Cross Provider (all)	Single Provider (7)	Cross Provider (7)	Cross Provider (all)
Centroid	232	166	170	760	456	574
Finger-printing	**94**	153	245	277	313	297
Gaussian Process	126	**87**	**65**	**196**	**147**	**134**

pattern is different on week days and week ends; (2) cellular signaling pattern is different at different times of a day; (3) Train users generate different cellular signaling pattern than car users; (4) in an event of incidence, the signaling notch (high decrease) followed by a peak occurs. This clearly indicates that cellular user classification using hand off or other cellular signaling data is possible.

Bar-Gera et al. [8] processed cellular data of handover events collected on 14km of Ayalon freeway in Israel with 10 interchanges in both directions during January-March 2005 to estimate travel times. The cellular based system received observations for about 1-3% of the total traffic during day time (1000-2000 hrs) and generated 63% valid travel time estimates for 27 road segments. Cellular data was more noisy (14%) than loop detector data (5%). The noise was measured as average absolute relative difference between travel time estimates for consecutive 5 min intervals. However the authors do not describe algorithms used for map matching and travel time estimation. Calabrese et al.[9] used cellular signaling information available at Abis interface (handover) and A-interface (location area updates) in the city of Rome. Abis signaling was processed in real time to predict Mobile Equipments(ME) position and speed to produce traffic map. Received signal power (RXLEV and RXQUAL) and Time Advance (TA) value were used to estimate location of active ME. Location error was reported to be 159 meters in urban area, 295 meters in suburban areas and up to 1457m in extra urban area. Error in considering moving ME as still and vice-verse was reported to be only 3.2%. Travel time estimation errors when compared to the readings taken by GPS and odometer, was 14.88% (bypass roads), 10.08% (primary urban streets), 17.66% (secondary urban streets). A-interface signaling was used to generate coercive grain location information about active or idle MEs using location area updates.

Traffic Online,Vodafone [10], analyzed signaling information on A-interface and A_{bis} interface to generate traffic information. They claim to generate high quality traffic information without mentioning methods/algorithms used in signal processing. Caceres et al.[11] observe that if the cell boundary or location area boundary is precisely known, the number of vehicles crossing that boundary can be counted just by counting number of hand overs (considering percentage of users making a call at given time). This can be used as an induction loop detector, counting number of vehicles crossing it, provided the cell is sectored and cell boundary or sector boundary maps to a unique road segment. However the assumptions made in the paper, i.e. cell boundary or location area boundary is precisely known and it maps to a unique road segment is very unrealistic. Cayford et al. at Institute of Transportation Studies, Berkeley conducted a study to evaluate effect of location accuracy, frequency of location measurements, and number of locations monitored, on traffic information generation using cellular network [12]. This section shows current GSM localization techniques.Current technology provides high accuracy, but requires substantial technological and financial investment.[13]

2.1.1 Cell-ID Based Technique

The basic position-finding technique used in GSM networks is based on cell identification.[4] This is a network-based method, where the cell that the handset is registered to is used to estimate the position of the user.[3] This technique does not use RSSI for estimate the location. This cell towers are the towers with the strongest RSSI.[2]

2.1.2 Fingerprinting Based Techniques

Fingerprinting based techniques stores the RSSI signatures at the different location in the database. This database is scanned for the closest area amid following stage in RSSI space to the unknown area. Fingerprints contains main cell ID,main cell RSSI, latitude and longitude and neighboring cell ID and RSSI.

Currently there are two types of technique for fingerprinting one is deterministic technique[5],[6] and other is probabilistic[2] technique. Deterministic methods does not consider the signal quality distribution. This technique use K-Nearest Neighbor classification algorithm.[5] During the online stage probabilistic technique store information about the RSSI dissemination in the fingerprint and try to estimate the location which has most probability. For instance, in Cellsense, the framework stores the RSSI information for each one cell ID at a specific area and uses Bayesian-based induction to predict the client area.[2]

Fingerprinting based procedure gives preferable precision over cell ID based method yet fingerprinting methods requires seeking a bigger database than cell-ID based techniques,overhead of building the unique finger impression is the same as developing the cell ID database as both oblige war driving.

2.1.3 CellSense Approach

CellSense works in two phase an offline stage and an online stage. In offline stage finger impression database is constructed.In finger impression database RSSI for each one cell ID is estimated. During the online tracking phase,the unique finger impression database is utilized to compute the likelihood of accepting the RSSI signal quality vector at the unknown area at every area in the unique finger impression.[2] The most likely area is utilized as the estimated areas. For prediction of the location they used Bayes' theorem.

2.1.4 Hidden Markov Model Based Localization

In [13] author propose a Hidden Markov Model based arrangement that influences the sign quality history from just the related cell tower to attain exact GSM limitation. They utilized just data of related cell tower in light of the fact that today numerous telephones with restricted API gives just related cell tower data.

2.1.5 Centroid Algorithm

This algorithm is quick to figure however doesn't utilize radio propagation model. This algorithm deals with lookup table which contain cell Id,latitude,longitude fields. The same cell ID is looked in lookup table,to get latitude and longitude data. In this manner the telephone's position can be dictated by geometric focal point of every cells. The issue with this algorithm is cell tower data is not given by administration supplier and we can't get the accurate position inside the cell.[5]

2.1.6 Monte Carlo Localization With Gaussian Process Signal Model

This algorithm utilize a radio propagation model and Markov localization to predict the phone's position.[5] Its construct a sensor model to anticipate the sign quality at every area. The phone's location is predicted using a Bayesian particle filter.[5] This methodology is similar to fingerprinting,but this methodology utilizes a dynamic parametric model of the sign environment whether in fingerprinting it is building an immediate search index of the adjustment information itself. In Monte-Carlo localization,the conviction about the telephone's position is spoken to by a situated of random samples and each one sample comprises of a state vector of the basic system,which is the position of the cellular telephone.[5]

We make the following observations about current state of related work: as per best of our knowledge, no study is carried out to assess cellular environment in India for its applicability to generate useful traffic information. Most of the papers in the literature describe their experimental/simulation results without providing details of algorithms used for generating traffic information. Their major focus is on using the cellular signaling mechanism to generate location information. Very little attempt is made in designing algorithms for generating useful traffic information using erroneous location estimates. Most of the papers do not consider large location error (more than 250 meters), which may be present in typical Indian scenario due to larger cell size.

Chapter 3

Implementation Methodology

Localization using GSM network works in two phase. In First phase our application records the GSM scenarios. In this phase GSM environment of multiple GSM network is concentrated and based on that our learning database is created. Other phase is prediction phase. In forecast phase we utilize database made as a part of past phase and tries to estimate the location. For prediction of the location we used cosine similarity function.

3.1 Cosine Similarity

The cosine similarity between two vectors is a measure that calculates the cosine of the point between them. This metric is a measurement of orientation and not magnitude,it can be seen as an examination between two vectors on a standardized space. For finding the cosine similarity equation we have to solve the equation of the dot product for the $\cos\theta$: [14]

$$\vec{a} \cdot \vec{b} = \| \vec{a} \| \| \vec{b} \| \cos \Theta$$
$$\cos \Theta = \vec{a} \cdot \vec{b} \div \| \vec{a} \| \| \vec{b} \|$$

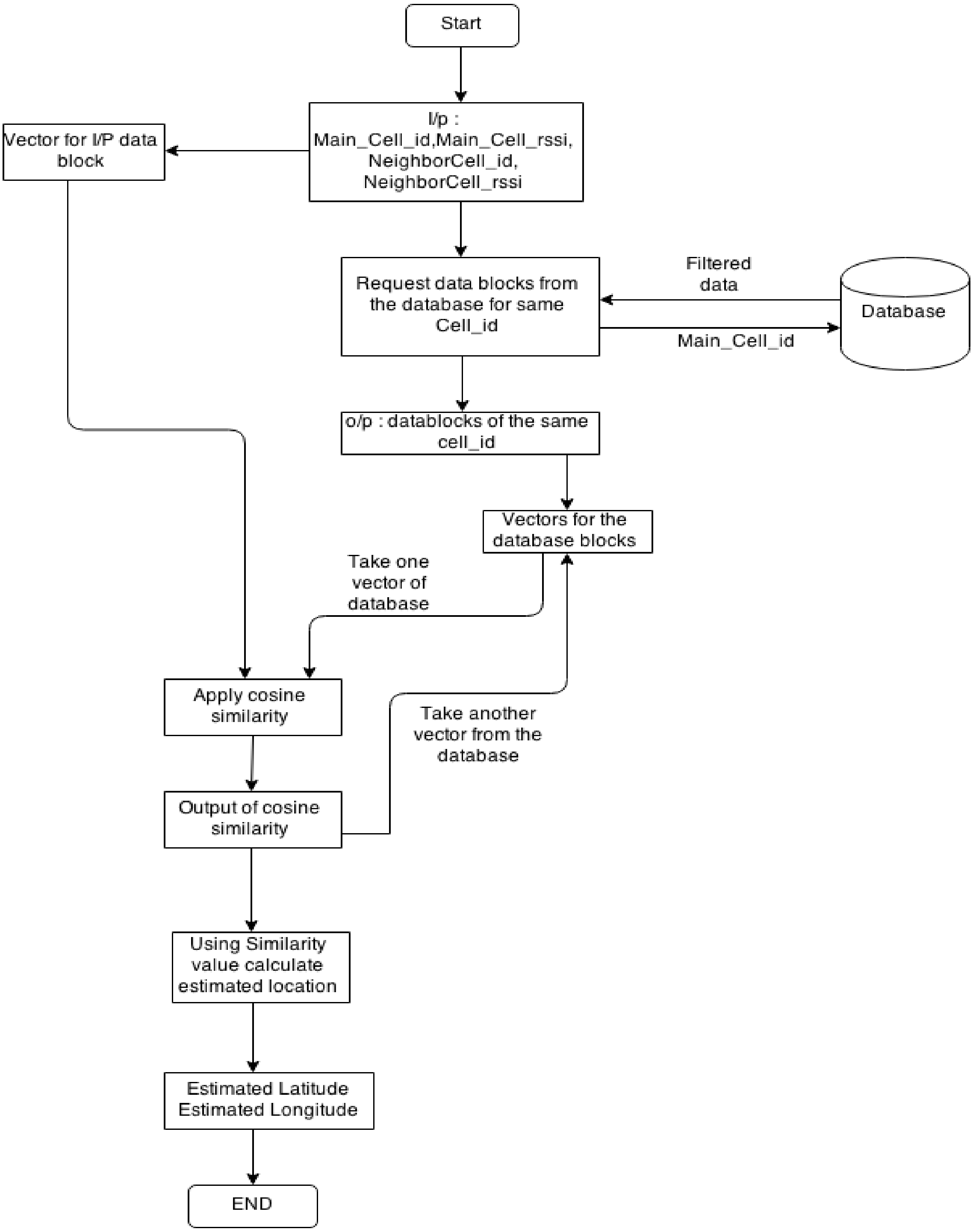
If the output of the above equation is 1 than both vectors are similar and if output is 0 than both vectors are dissimilar.

3.1.1 Proposed Algorithm Using Cosine Similarity

This algorithm has two phase one is learning phase and other is prediction phase. In learning phase we created database using our android application. For learning database we have to drive with android phone and run the application at different places of Ahmadabad. We are created database by taking readings nine or ten times at same places. We proposed new algorithm for anticipate the location just by taking GSM traces as input. For prediction we utilized database made in the previous phase. This database contains the main cell ID,main cell RSSI,Latitude,Longitude,neighbor cell ID,neighbor cell RSSI. This database is made amid the learning process accordingly more exact learning will be needed for exact database. To this database we will give main cell ID,main cell RSSI,neighbor cell IDs,neighbor cell RSSI as input.

Now we will request our database to filter data blocks which have same main cell ID as input main cell ID. So database gives only filtered data blocks which have same main cell ID as input main cell ID. Now we will represent input data block as a vector and also represent one data block from the filtered database as vector. After that we will apply cosine similarity function two both vector. Now take another data block from the filtered database and continue this process. After this process result of cosine similarity is checked that which data block gives result nearly 1 is the most probable similar data block to the input data block. So extract the Latitude and Longitude from that data block as the output. After that using estimated and actual latitudes and longitudes we can find distance between two points. Figure 3.1 shows the flowchart of the algorithm.

Figure 3.1: Flowchart of algorithm for location prediction using cosine similarity



3.1.2 Algorithm

Algorithm 1 Cosine Similarity For Localization Algorithm

procedure MAIN
  Give n input blocks to algorithm which contains Main-Cell id, Main-Cell Rssi, Neighbor-Cell ids(c1,c2...c6) and Neighbor-Cell Rssi(r1,r2,...r6) values.
  Filter learning database using Main-Cell id. Now filtered database contains m blocks which contains Main-Cell Rssi, Neighbor-Cell ids(C1,C2...C6) and Neighbor-Cell Rssi(R1,R2,...R6), Longitude(log), Latitude(lat) values.
  Consider input blocks and filtered database blocks as a vector.
  Apply cosine similarity between this vectors and estimate the location as follows:
  for $(i = 0; i < n; i++)$ **do**
    for $(j = 0; i < m; j++)$ **do**
      $COSINE - SIMILARITY()$
    end for
  end for
  $ESTIMATE - LOCATION()$
end procedure

Algorithm 2 COSINE-SIMILARITY function

procedure COSINE-SIMILARITY
  for $i = 0; i < 6; i++$ **do**
    if $Ci == ci$ **then**
      similarity(S(k)) $= \sum \frac{ri*Ri}{\sqrt{\sum_0^5 ri^2}*\sqrt{\sum_0^5 Ri^2}}$
    end if
  end for
  Return $S(k)$
end procedure

Algorithm 3 ESTIMATE-LOCATION function

procedure ESTIMATE-LOCATION
    Map latitude and longitude with corresponding similarity value in list and sort the list using similarity value.
    k=5
    for $i = 0; i < k; i++$ **do**
        est-latitude $= \sum \frac{S(i)*lat(i)}{\sum_0^k S(k)}$
        est-longitude $= \sum \frac{S(i)*log(i)}{\sum_0^k S(k)}$
    end for
    print est-latitude
    print est-longitude
end procedure

3.2 Chi-Square Goodness-Of-Fit-Test

The Chi-square goodness-of-fit test is used to see if the frequency distribution fits a specific pattern. In this test there are two values one is observed value and one is expected value. Observed value is the frequency of a category from the samples and expected value is the frequency of the input samples. Using following equation we can find chi-square value.

$$\chi^2 = \sum \frac{(observed - expected)^2}{expected}$$

3.2.1 Proposed Algorithm Using Chi-square Similarity

This algorithm has two phase one is learning phase and other is prediction phase. In learning phase we are created database using our android application. For learning database we have to drive with android phone and run the application at different places of Ahmadabad. We are creating database by taking readings nine or ten times at same places. We proposed new algorithm for anticipate the location just by taking GSM traces as input. For prediction we utilized database made in the previous phase. This database contains the main cell ID,main cell RSSI,Latitude,Longitude,neighbor cell ID,neighbor cell RSSI. This database is made amid the learning process accordingly more exact learning will be needed for exact database. To this database we will give main cell ID,main cell RSSI,neighbor cell IDs,neighbor cell RSSI as input.

Now we will request our database to give data blocks which have same main cell ID as input main cell ID. So database will give only filtered data blocks which have same main cell ID as input main cell ID. Now consider one data block from the input as a expected frequency and one data block from the database as a observed frequency. After that we will apply chi-square similarity function and find the value of chi-square and Degree of freedom. Using this chi-square and degree of freedom values we will calculate p-value. now using this p-value we will predict the location and find the estimated latitude and estimated longitude. After that using estimated and actual latitudes and longitudes we can find distance between two points. Figure 3.2 shows the flow chart of the algorithm.

3.2.2 Algorithm

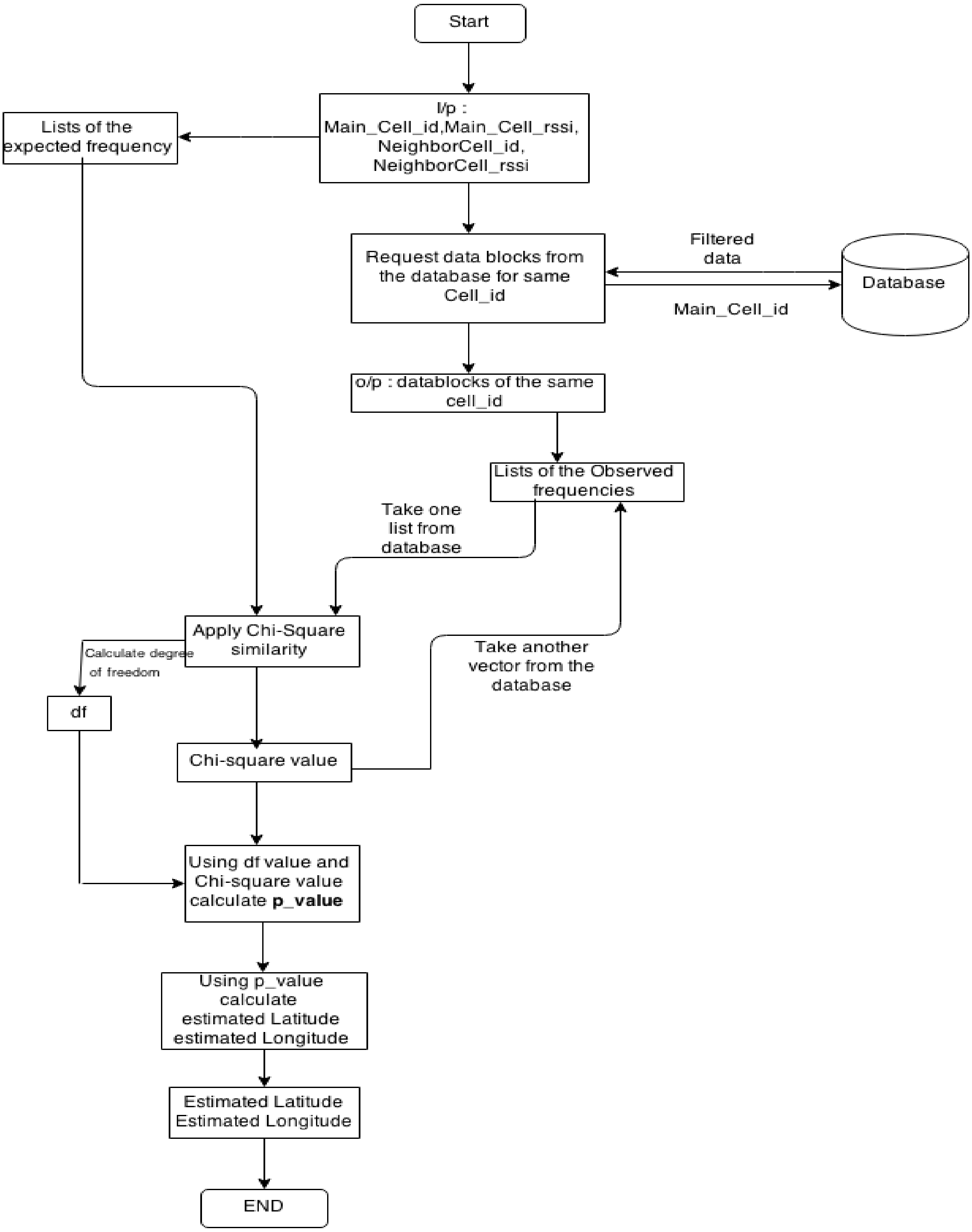
Algorithm 4 Chi-square Similarity For Localization Algorithm

procedure MAIN
    Give n input blocks to algorithm which contains Main-Cell id, Main-Cell Rssi, Neighbor-Cell ids(c1,c2...c6) and Neighbor-Cell Rssi(r1,r2,...r6) values.
    Filter learning database using Main-Cell id. Now filtered database contains m blocks which contains Main-Cell Rssi, Neighbor-Cell ids(C1,C2...C6) and Neighbor-Cell Rssi(R1,R2,...R6), Longitude(log), Latitude(lat) values.
    Consider input blocks and filtered database blocks as a expected and observed frequencies.
    Apply chi-square goodness-of-fit test for similarity between this frequencies and estimate the location as follows :
    for $(i = 0; i < n; i++)$ **do**
        for $(j = 0; i < m; j++)$ **do**
            $CHISQUARE - SIMILARITY()$
        end for
    end for
    $ESTIMATE - LOCATION()$
end procedure

Algorithm 5 CHI-SQUARE-SIMILARITY function

procedure CHI-SQUARE-SIMILARITY
    for $i = 0; i < 6; i++$ **do**
        if $Ci == ci$ **then**
            chi-square(S(k)) $= \sum \frac{((observed[i]) - (expected[i]))^2}{expected[i]}$
        end if
    end for
    Return $S(k)$
    Calculate p-value(P) using chi-square and DF value.
end procedure

Figure 3.2: Flowchart of algorithm for location prediction using chi-square similarity



Algorithm 6 ESTIMATE-LOCATION function

```
procedure ESTIMATE-LOCATION
    Map latitude and longitude with corresponding p-value(P) in list and sort the list using similarity value.
    k=5
    for i = 0; i < k; i++ do
        est-latitude = Σ (P(i)*lat(i)) / (Σ_0^k P(k))
        est-longitude = Σ (P(i)*log(i)) / (Σ_0^k P(k))
    end for
    print est-latitude
    print est-longitude
end procedure
```

Chapter 4

Experiment

4.1 Application Development

In our project we are collecting the data of cellular network and also collecting the GPS information.For collecting this information we made an android application.This application is compatible to android version 2.2 or greater.This application can get following information of every second.

- MCC(Mobile Country Code)
- MNC(Mobile Network Code)
- Main Cell-Id
- Neighbor Cell-Id
- Signal Strength
- Latitude
- Longitude
- Date And Time

For further analysis of the output we have to store the output.We created application such that we can store the output of every second.For run this application we use Android phone which will allow us to give all of above information.Following figure 4.1 shows the snapshots of the android application.

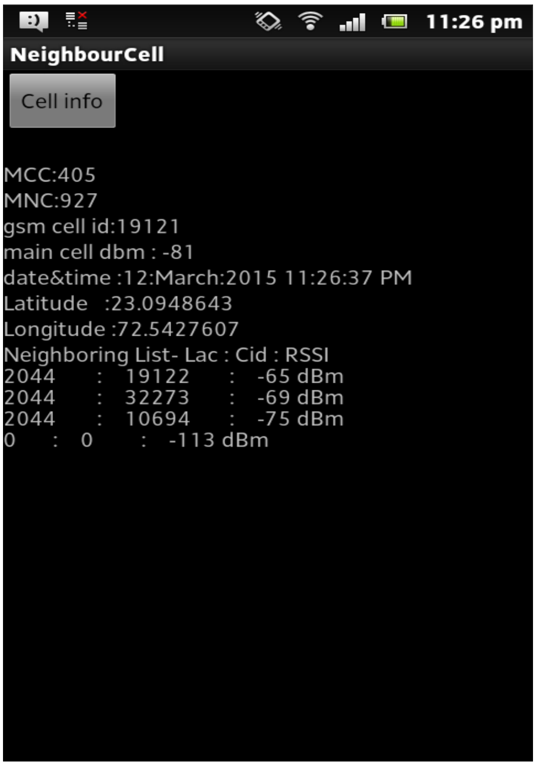


Figure 4.1: Screen shot of the application

Figure 4.2 is snapshot of the file which is made by the application.Thus using this application we can record cell ID and RSSI fingerprint for cellular networks at different location.

4.2 Data Collection

To gain some insight into cellular environment in Indian Urban area and to see whether vehicle tracking is feasible or not on Indian road network, we carried out the following experiment:We installed our app in sony erricsion android phone which has android version 4.2 and carried it while moving from Bapunagar area to Nirma university in Ahmedabad city of Gujarat.The movement stretch is 17 kilometers long and traverses through typical urban roads of Indian city.For creation of learning database we took the traces of this single route for 10 days.After taking 10 days traces prepossessing of that traces has to be done.For that we remove traces which have zero neighbor cell and also remove traces which have errors in latitude and longitude.This 10 days traces are stored in text file so we converted it into xml file using python programming language.Using this xml files we created our learning database.Below figure 4.3 shows the route for the data collection.

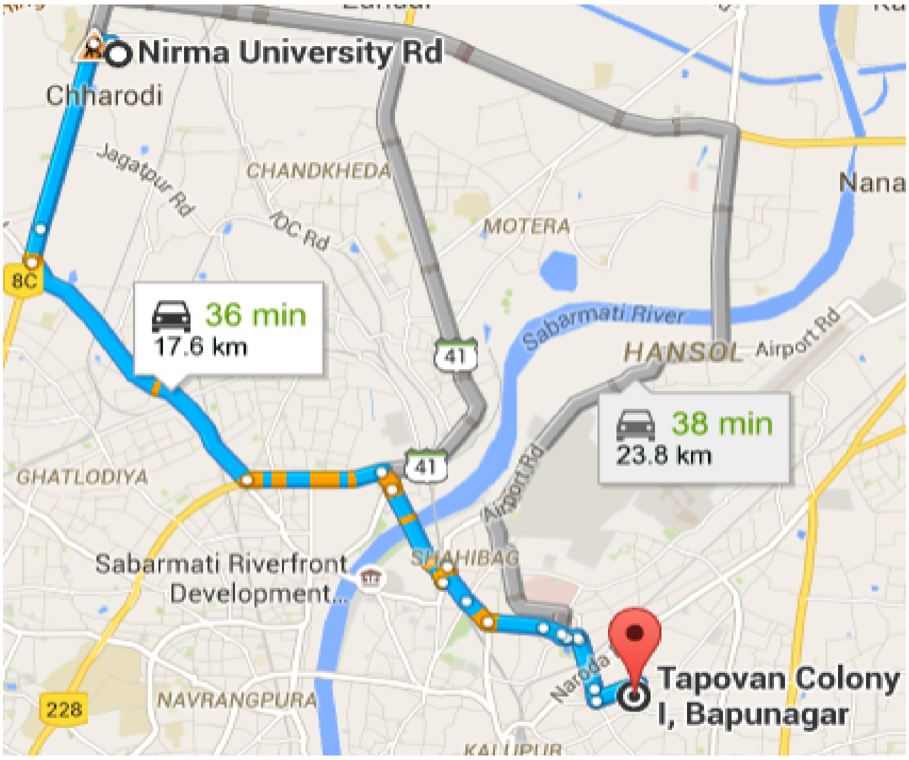
Figure 4.2: Readings of Android Application

```
date&time :25:February:2015 04:02:44 PM
main cell dbm : -91
Neighboring List- Lac : Cid : RSSI
3002      :    10211      :    -79 dBm
3002      :    10233      :    -79 dBm

MCC:405
MCC:802
gsm cell id:19737561
Latitude   :23.11541666666667
Longitude :72.53663499999999
date&time :25:February:2015 04:02:45 PM
main cell dbm : -91
Neighboring List- Lac : Cid : RSSI
3002      :    10211      :    -77 dBm
3002      :    17243      :    -81 dBm
3002      :    10233      :    -83 dBm
3002      :    10232      :    -89 dBm

MCC:405
MCC:802
gsm cell id:19737561
Latitude   :23.11541666666667
Longitude :72.53663499999999
date&time :25:February:2015 04:02:46 PM
main cell dbm : -91
Neighboring List- Lac : Cid : RSSI
3002      :    10211      :    -77 dBm
3002      :    17243      :    -81 dBm
3002      :    10233      :    -83 dBm
3002      :    10232      :    -89 dBm
```

Figure 4.3: Route For Data Collection



4.3 Experiment Result Analysis

We implemented our algorithms in python programming language. Traces from 10 days we take traces of 9 days as learning database and use one day traces as input for our algorithms and repeat this process by considered alternative days as input. Using our algorithms we predict the estimated latitude and estimated longitude and also calculate the error between actual and estimated location. We also found average error,median error and ninety percent error for every 30 seconds interval for traces for every day. We also tried to find relation between average error and average number of input neighbor cells,average number of learning neighbor cells,average input rssi and average learning rssi. We generated following four graphs for the all days considered as input for both algorithms.

- Average Input Cell vs Average Error
- Average Input Rssi vs Average Error
- Average Learning Cell vs Average Error
- Average Learning Rssi vs Average Error

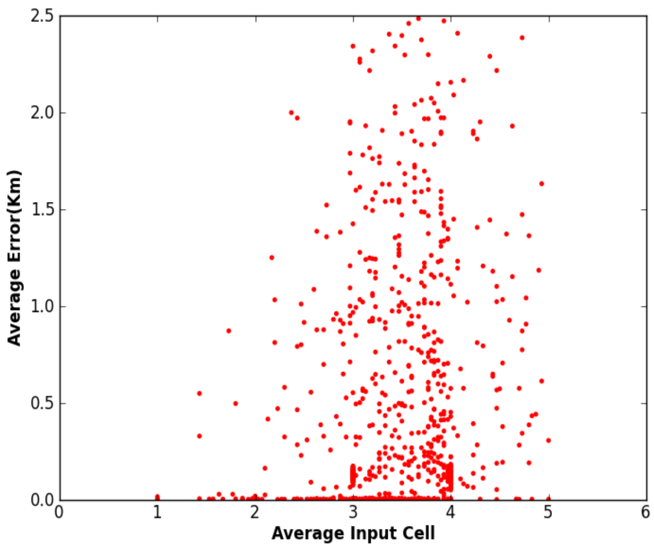
Table 4.1: Correlation Coefficient Using Chi-square Similarity

Parameter	Average Error
Average Input Cell	0.21795491
Average Input Rssi	0.42105443
Average Learning Cell	0.29677661
Average Learning Rssi	0.40096255

4.3.1 Chi-Square Goodness-Of-Fit-Test Results

Now We merged all days data and generated this four graphs which is shown in below figures.We also calculated correlation coefficient for this graphs it is as shown in table 4.1.

Figure 4.4: Average Input Cell vs Average Error



4.3.2 Cosine-Similarity Results

We repeated the same process and merge the data of all days as we did in above algorithm and generate the graphs as shown in figures below and also calculated correlation coefficient for all the parameters shown in table 4.2.

Figure 4.5: Average Input Rssi vs Average Error

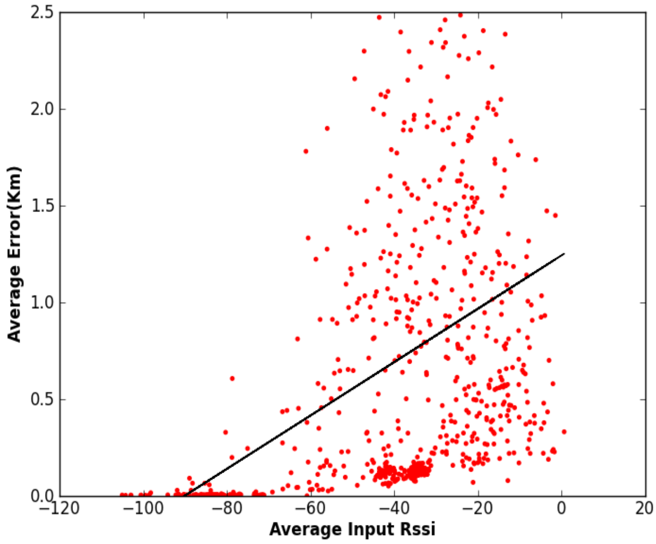


Figure 4.6: Average Learning Cell vs Average Error

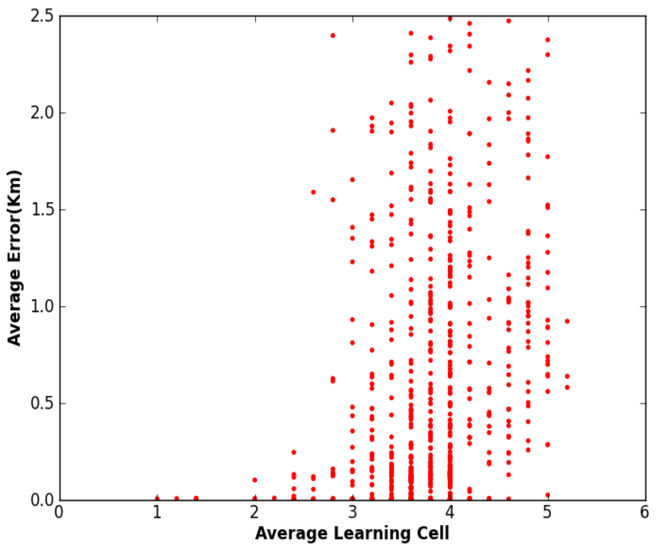


Figure 4.7: Average Learning Rssi vs Average Error

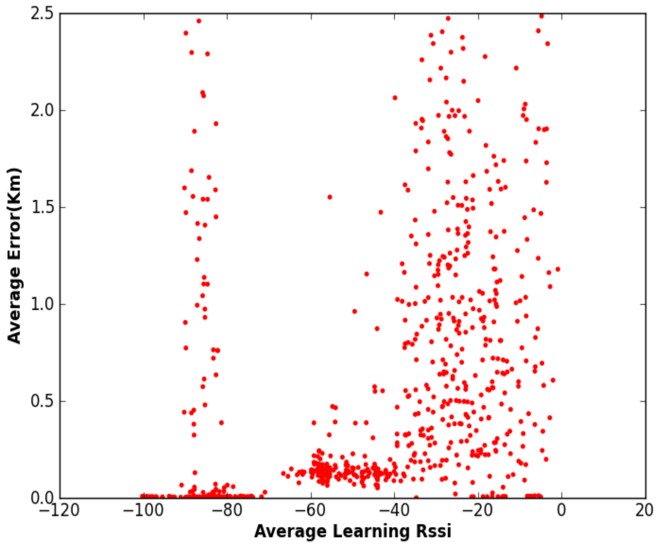


Table 4.2: Correlation Coefficient Using Cosine-similarity

Parameter	Average Error
Average Input Cell	0.31148516
Average Input Rssi	0.42280925
Average Learning Cell	0.1388458
Average Learning Rssi	0.19676975

Figure 4.8: Average Input Cell vs Average Error

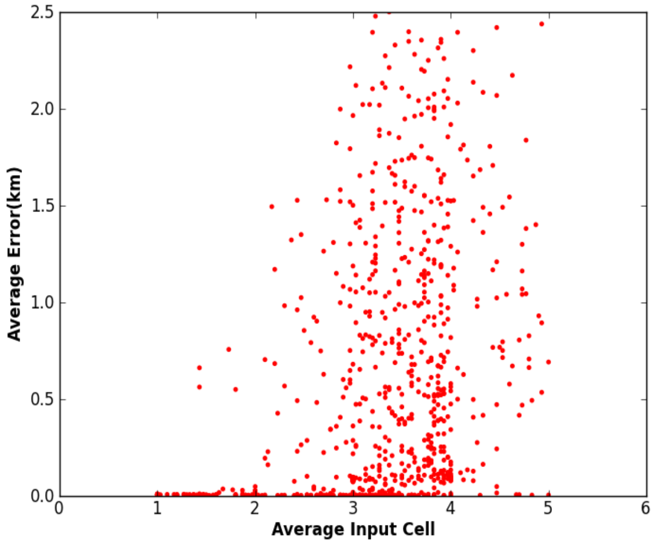


Figure 4.9: Average Input Rssi vs Average Error

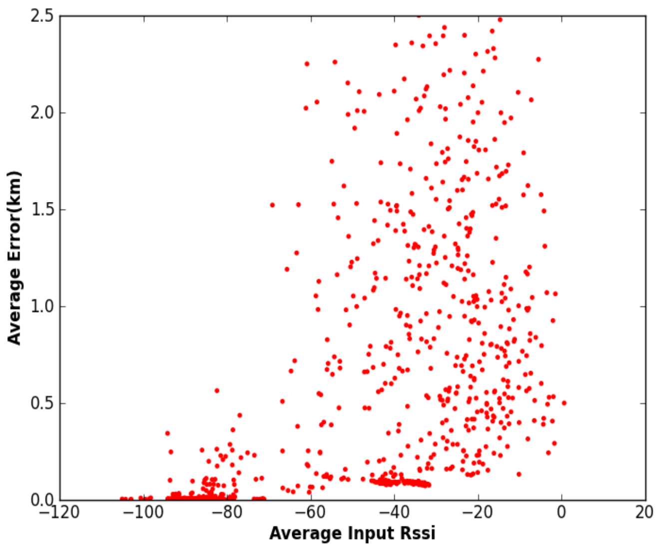


Figure 4.10: Average Learning Cell vs Average Error

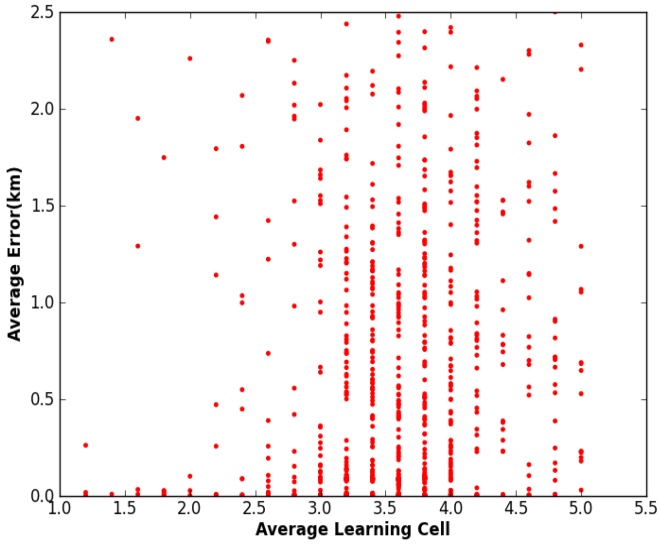
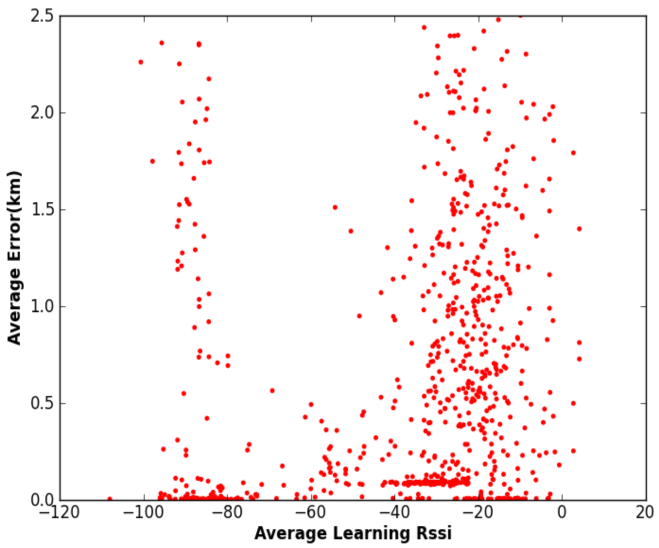


Figure 4.11: Average Learning Rssi vs Average Error



Chapter 5

Conclusion and Future Plan

5.1 Conclusion

We started our experiment to get the alternative of GPS technology. For that we developed an android application for storing the GSM and GPS traces of multiple GSM network and developed the algorithm which may be gives accurate location. Using this reading we created a learning database for the multiple cellular network. We implemented two different algorithm for location prediction. We analyzed the result of experiment. We tried to find relation between average error and average number of cells and average rssi values but we did not get any trend in our graphs,So we can say that cellular infrastructure has a random behavior.

5.2 Future Work

In future,We will create learning database for different areas and analyze the result of different areas. And also try to develop android application which will use this learning database and predict a location using our algorithm.

References

[1] "Press release no. 32/2015," Telecom Regulatory Authority of India, 2015.

[2] M. tech Systems, "Cellsense: A probabilistic rssi-based gsm positioning system," IEEE Globecom 2010 proceedings, 2010.

[3] A. V. Emiliano Trevisani, "Cell-id location technique, limits and benefits: an experimental study," Mobile Computing Systems and Applications, 2004.

[4] W. B. a. J. M. K. Raja, "We know where you are," IEE Communication Engineer, July, 2006.

[5] D. C. A. V. Mike Y. Chen, Timothy Sohn, "Practical metropolitan-scale positioning for gsm phones," September 17-21, 2006.

[6] F. P. K. T. Alex Varshavsky, MikeY. Chen, "Are gsm phones the solution for localization?," Mobile Computing Systems and Applications. WMCSA '06. Proceedings. 7th IEEE Workshop on, 2006.

[7] F. R. D. Valerio, A. DAlconzo and W. Wiedermann, "Exploiting cellular networks for road traffic estimation: a survey and a research road map," Vehicular Traffic Conference, 2009.

[8] H. Bar-Gera, "Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from israel," Transportation Research Part C 15, Elsevier, 2007.

[9] P. L. D. P. F. Calabrese, M. Colonna and C. Ratti, "Realtime urban monitoring using cellular phones: a case-study in rome," IEEE Transactions on Intelligent Transportation Systems, 2010.

[10] C. Birle, “Traffic online via gsm network data to traffic information in real time,,” Presentation Vodafone, 2007.

[11] J. W. N. Caceres and F. Benitez, “Deriving origin-destination data from a mobile phone network,” The Institution of Engineering and Technology Intelligent Transport System, 2007.

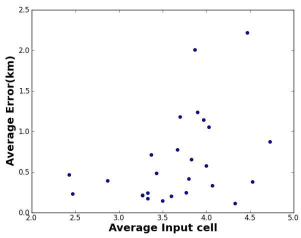
[12] R. Cayford and T. Johnson, “Operational parameters affecting the use of anonymous cell phone tracking for generating traffic information,” TRB Annual Meeting, 2003.

[13] M. tech Systems, “A hidden markov model for localization using low-end gsm cell phones,” IEEE ICC 2011 proceedings, 2011.

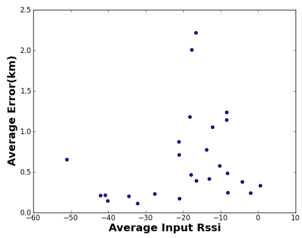
[14] “http://pyevolve.sourceforge.net/wordpress/?p=2497.”

Appendix A

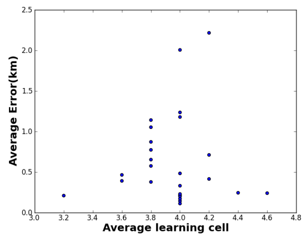
Localization Using Chi-Square Similarity



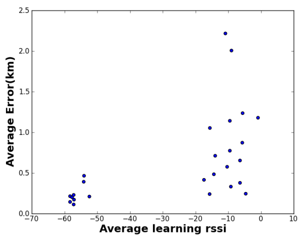
(a) Avg Input Cell vs Avg Error(Day1)



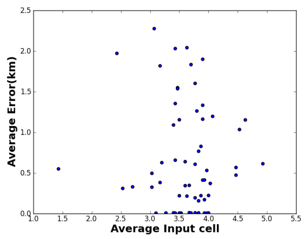
(b) Avg Input Rssi vs Avg Error(Day1)



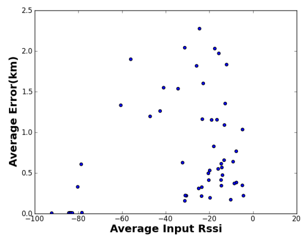
(c) Avg Learning Cell vs Avg Error(Day1)



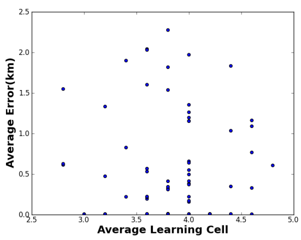
(d) Avg Learning Rssi vs Avg Error(Day1)



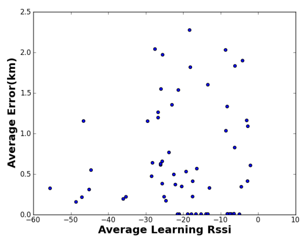
(e) Avg Input Cell vs Avg Error(Day2)



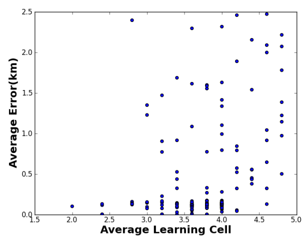
(f) Avg Input Rssi vs Avg Error(Day2)



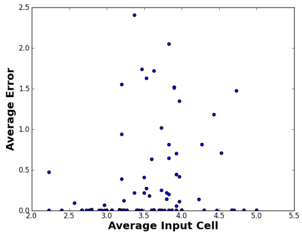
(g) Avg Learning Cell vs Avg Error(Day2)



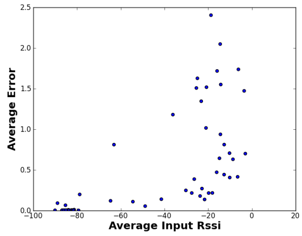
(h) Avg Learning Rssi vs Avg Error(Day2)



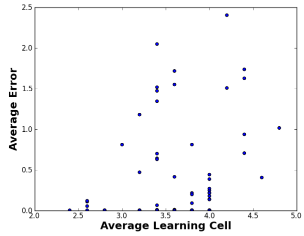
(i) Avg Learning Cell vs Avg Error(Day10)



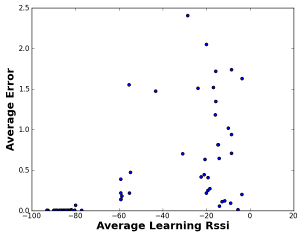
(a) Avg Input Cell vs Avg Error(Day3)



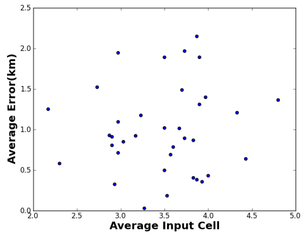
(b) Avg Input Rssi vs Avg Error(Day3)



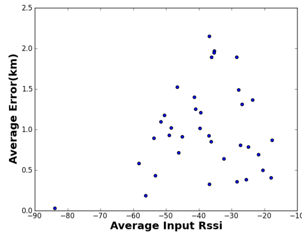
(c) Avg Learning Cell vs Avg Error(Day3)



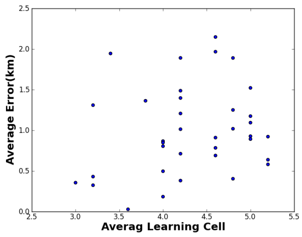
(d) Avg Learning Rssi vs Avg Error(Day3)



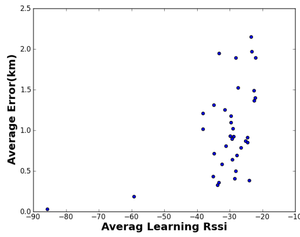
(e) Avg Input Cell vs Avg Error(Day4)



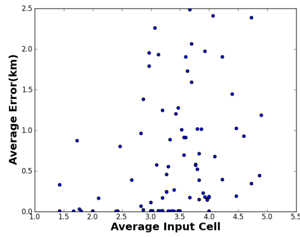
(f) Avg Input Rssi vs Avg Error(Day4)



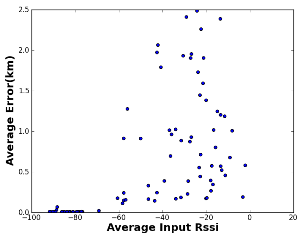
(g) Avg Learning Cell vs Avg Error(Day4)



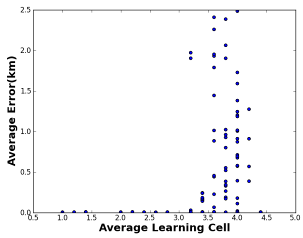
(h) Avg Learning Rssi vs Avg Error(Day4)



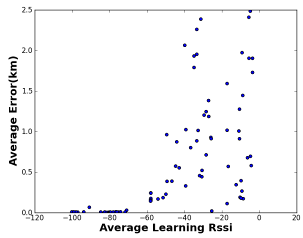
(i) Avg Input Cell vs Avg Error(Day5)



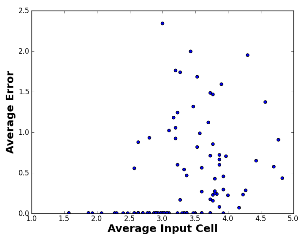
(j) Avg Input Rssi vs Avg Error(Day5)



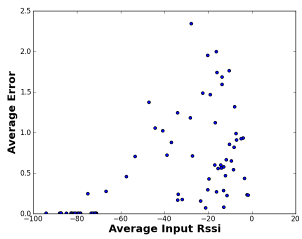
(k) Avg Learning Cell vs Avg Error(Day5)



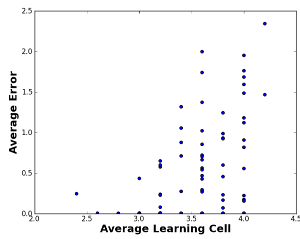
(l) Avg Learning Rssi vs Avg Error(Day5)



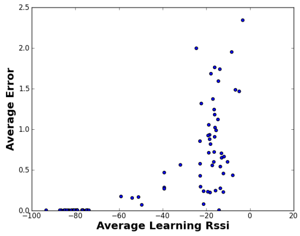
(m) Avg Input Cell vs Avg Error(Day6)



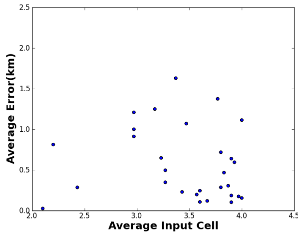
(n) Avg Input Rssi vs Avg Error(Day6)



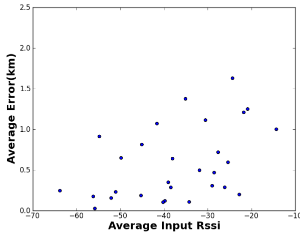
(o) Avg Learning Cell vs Avg Error(Day6)



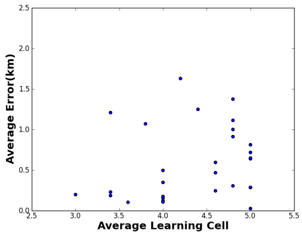
(a) Avg Learning Rssi vs Avg Error(Day6)



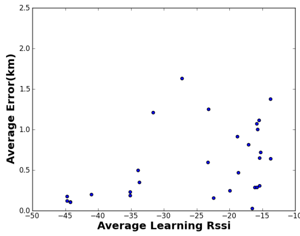
(b) Avg Input Cell vs Avg Error(Day7)



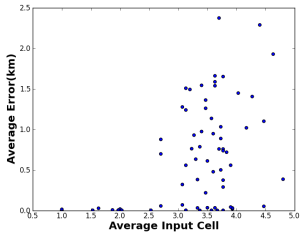
(c) Avg Input Rssi vs Avg Error(Day7)



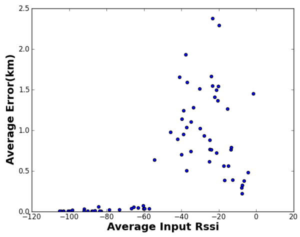
(d) Avg Learning Cell vs Avg Error(Day7)



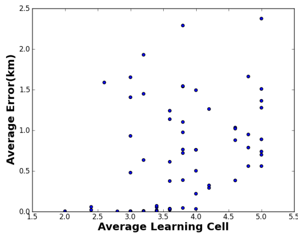
(e) Avg Learning Rssi vs Avg Error(Day7)



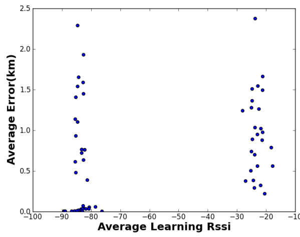
(f) Avg Input Cell vs Avg Error(Day8)



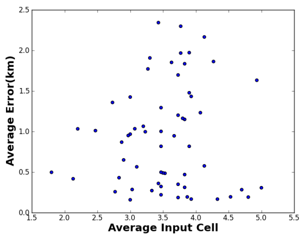
(g) Avg Input Rssi vs Avg Error(Day8)



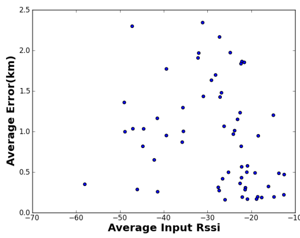
(h) Avg Learning Cell vs Avg Error(Day8)



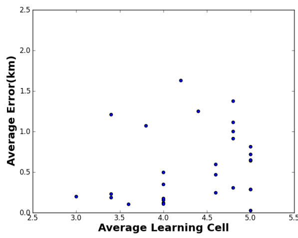
(i) Avg Learning Rssi vs Avg Error(Day8)



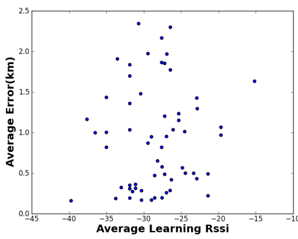
(j) Avg Input Cell vs Avg Error(Day9)



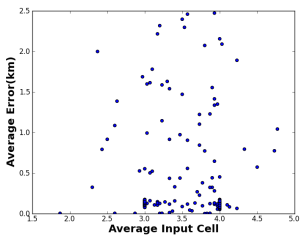
(k) Avg Input Rssi vs Avg Error(Day9)



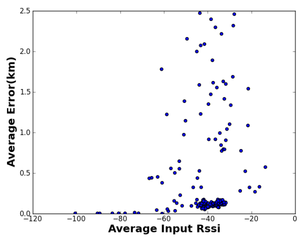
(l) Avg Learning Cell vs Avg Error(Day9)



(m) Avg Learning Rssi vs Avg Error(Day9)



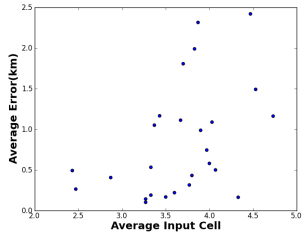
(n) Avg Input Cell vs Avg Error(Day10)



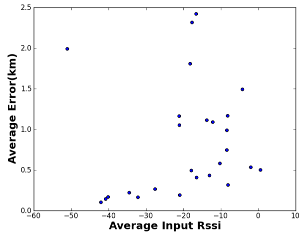
(o) Avg Input Rssi vs Avg Error(Day10)

Appendix B

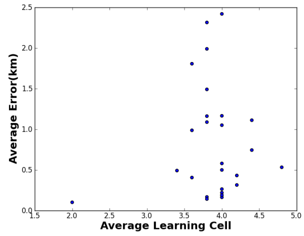
Localization Using Cosine Similarity



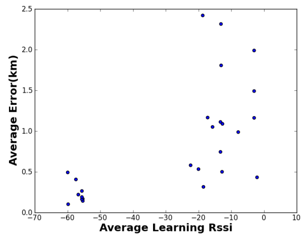
(a) Avg Input Cell vs Avg Error(Day1)



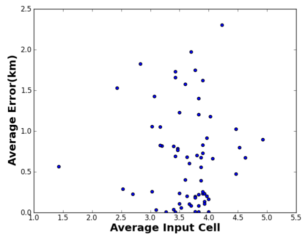
(b) Avg Input Rssi vs Avg Error(Day1)



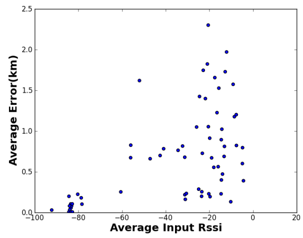
(c) Avg Learning Cell vs Avg Error(Day1)



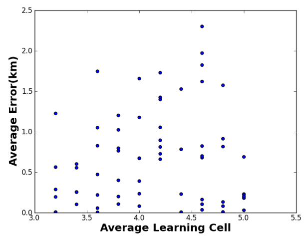
(d) Avg Learning Rssi vs Avg Error(Day1)



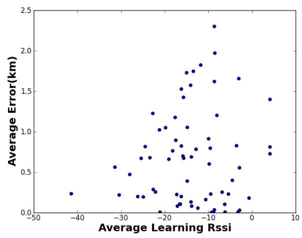
(e) Avg Input Cell vs Avg Error(Day2)



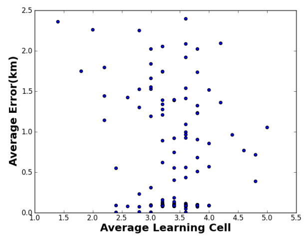
(f) Avg Input Rssi vs Avg Error(Day2)



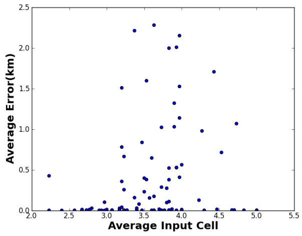
(g) Avg Learning Cell vs Avg Error(Day2)



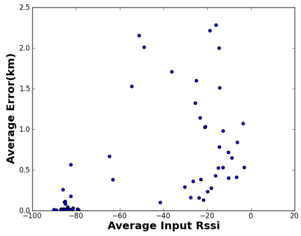
(h) Avg Learning Rssi vs Avg Error(Day2)



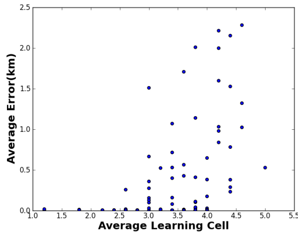
(i) Avg Learning Cell vs Avg Error(Day10)



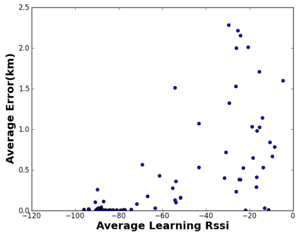
(a) Avg Input Cell vs Avg Error(Day3)



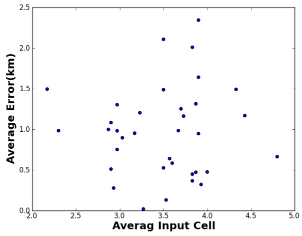
(b) Avg Input Rssi vs Avg Error(Day3)



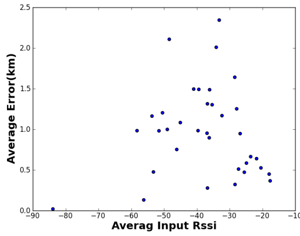
(c) Avg Learning Cell vs Avg Error(Day3)



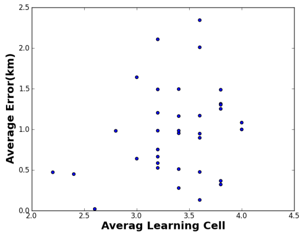
(d) Avg Learning Rssi vs Avg Error(Day3)



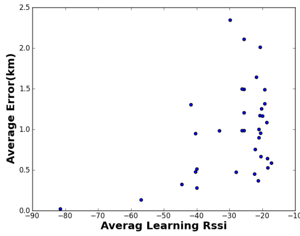
(e) Avg Input Cell vs Avg Error(Day4)



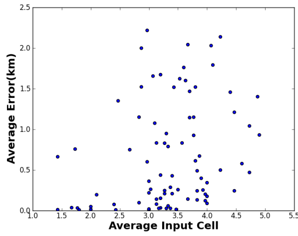
(f) Avg Input Rssi vs Avg Error(Day4)



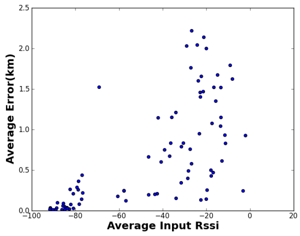
(g) Avg Learning Cell vs Avg Error(Day4)



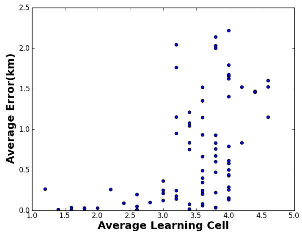
(h) Avg Learning Rssi vs Avg Error(Day4)



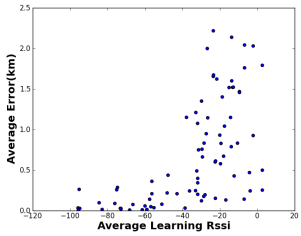
(i) Avg Input Cell vs Avg Error(Day5)



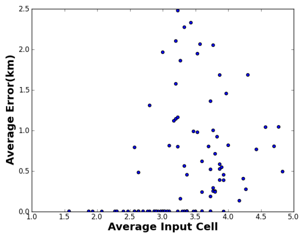
(j) Avg Input Rssi vs Avg Error(Day5)



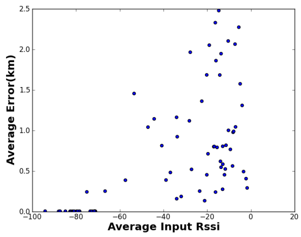
(k) Avg Learning Cell vs Avg Error(Day5)



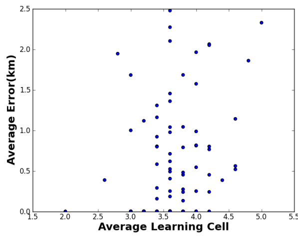
(l) Avg Learning Rssi vs Avg Error(Day5)



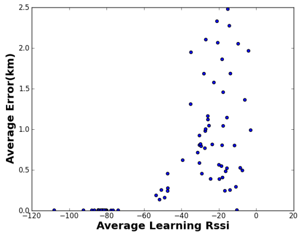
(m) Avg Input Cell vs Avg Error(Day6)



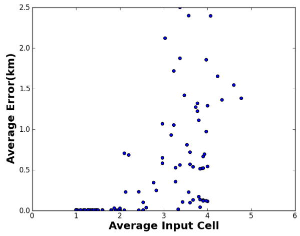
(n) Avg Input Rssi vs Avg Error(Day6)



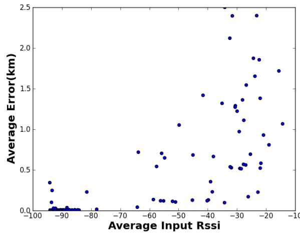
(o) Avg Learning Cell vs Avg Error(Day6)



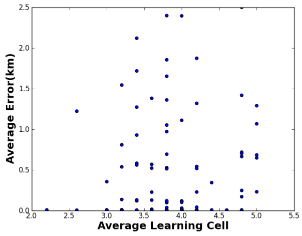
(a) Avg Learning Rssi vs Avg Error(Day6)



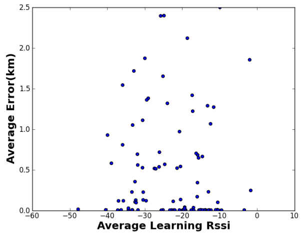
(b) Avg Input Cell vs Avg Error(Day7)



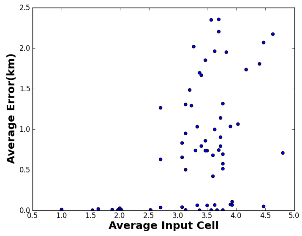
(c) Avg Input Rssi vs Avg Error(Day7)



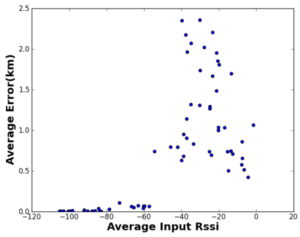
(d) Avg Learning Cell vs Avg Error(Day7)



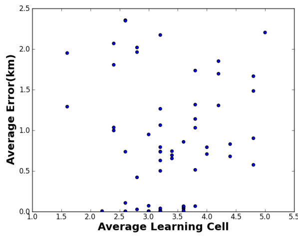
(e) Avg Learning Rssi vs Avg Error(Day7)



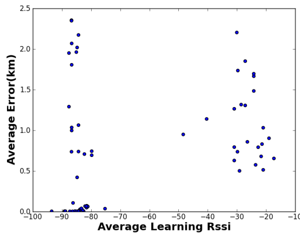
(f) Avg Input Cell vs Avg Error(Day8)



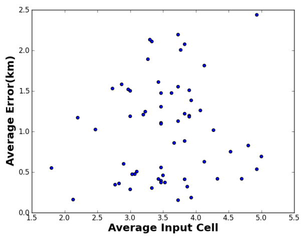
(g) Avg Input Rssi vs Avg Error(Day8)



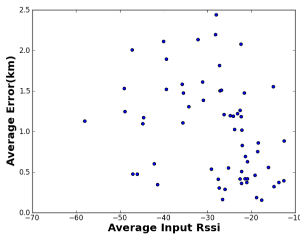
(h) Avg Learning Cell vs Avg Error(Day8)



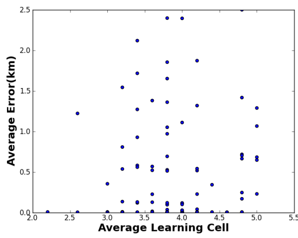
(i) Avg Learning Rssi vs Avg Error(Day8)



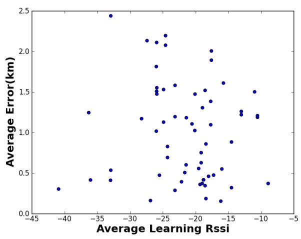
(j) Avg Input Cell vs Avg Error(Day9)



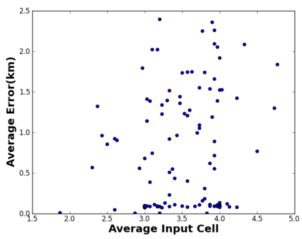
(k) Avg Input Rssi vs Avg Error(Day9)



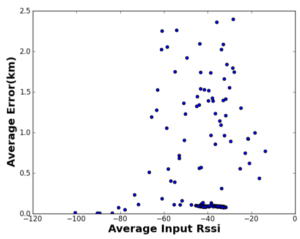
(l) Avg Learning Cell vs Avg Error(Day9)



(m) Avg Learning Rssi vs Avg Error(Day9)



(n) Avg Input Cell vs Avg Error(Day10)



(o) Avg Input Rssi vs Avg Error(Day10)